

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-227320

(43) 公開日 平成8年(1996)9月3日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 D 3/12	3 0 5		G 0 5 D 3/12	3 0 5 V
B 2 5 J 9/16			B 2 5 J 9/16	
G 0 5 B 11/36	5 0 1	7531-3H	G 0 5 B 11/36	5 0 1 Q
	5 0 5	7531-3H		5 0 5 B

審査請求 未請求 請求項の数 5 F D (全 10 頁)

(21) 出願番号 特願平7-55189

(22) 出願日 平成7年(1995)2月21日

(71) 出願人 390008235

ファナック株式会社

山梨県南都留郡忍野村忍草字古馬場3580番地

(72) 発明者 加藤 哲朗

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

(72) 発明者 土田 行信

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

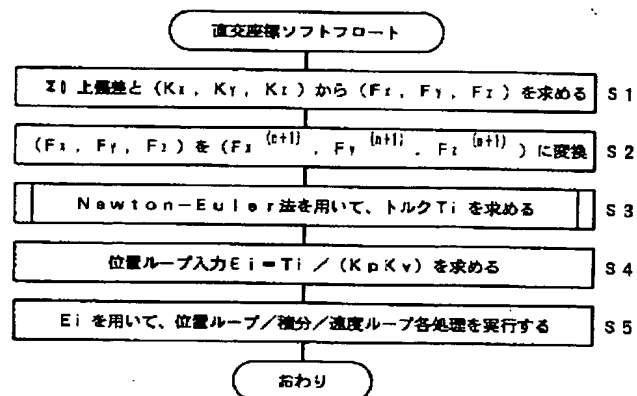
(74) 代理人 弁理士 竹本 松司 (外4名)

(54) 【発明の名称】 直交座標系上で柔らかさが調節可能なサーボ系

(57) 【要約】

【目的】 直交座標系上でソフトフロート機能をロボットに調節すること。

【構成】 直交座標系 $\Sigma 0$ 上における位置偏差と設定されたパラメータ (K_x , K_y , K_z) を用い、直交座標系 $\Sigma 0$ 上における力 (F_x , F_y , F_z) を求める。ステップ S 2 で、求められた力 (F_x , F_y , F_z) をロボットの姿勢データを用いてツール座標系 Σ_{n+1} 上のデータ ($F_x^{(n+1)}$, $F_y^{(n+1)}$, $F_z^{(n+1)}$) に変換する。更に、($F_x^{(n+1)}$, $F_y^{(n+1)}$, $F_z^{(n+1)}$) = ($f_x^{(n+1)}$, $f_y^{(n+1)}$, $f_z^{(n+1)}$) として、Newton-Euler 法により、各軸のトルク T_i を計算し (ステップ S 3)、 $T_i / (K_p K_v)$ から位置ループ入力 E_i を計算する (ステップ S 4)。位置ループ入力 E_i を入力とする位置ループ処理、位置ループ出力を入力とする積分処理と速度ループ処理、積分処理と速度ループ処理の出力を加算するトルク指令作成処理等を実行すれば (ステップ S 5)、ソフトフロート機能が発揮される。積分処理と速度ループ処理の出力の加算は、設定された制限範囲を越えないように行なわれる。



【特許請求の範囲】

【請求項 1】 ロボットの各軸を駆動するサーボモータを制御する為に前記各軸毎に位置ループ処理と速度ループ処理を含む処理を行なうサーボ系であって、

作業空間内に定義された直交座標系上での前記サーボの柔らかさを表わすパラメータの設定値の組と、前記直交座標系上での位置偏差から計算される前記直交座標系上での力を逆動力学で解いて求められた各軸のトルク T_i ($i = 1, 2 \dots n$; 但し、 n はロボットの軸数)

と、前記位置ループ処理のゲイン K_p と、速度ループ処理のゲイン K_v に基づいて算出された E_i を、各軸の前記位置ループ処理の入力とした、直交座標系上で柔らかさが調節可能な前記サーボ系。

【請求項 2】 ロボットの各軸を駆動するサーボモータを制御する為に前記各軸毎に位置ループ処理と速度ループ処理を含む処理を行なうサーボ系であって、

作業空間内に定義された直交座標系上での前記サーボの柔らかさを表わすパラメータの設定値の組と前記直交座標系上での位置偏差から計算される前記直交座標系上での力を逆動力学で解いて求められた各軸のトルク T_i ($i = 1, 2 \dots n$; 但し、 n はロボットの軸数) を

前記位置ループ処理のゲイン K_p と速度ループ処理のゲイン K_v の積で除した値 $E_i = T_i / (K_p K_v)$ を、各軸の前記位置ループ処理の入力とした、直交座標系上で柔らかさが調節可能な前記サーボ系。

【請求項 3】 ロボットの各軸を駆動するサーボモータを制御する為に前記各軸毎に位置ループ処理と速度ループ処理を含む処理を行なうサーボ系であって、

作業空間内に定義された直交座標系上での前記サーボの柔らかさを表わすパラメータの設定値の組と前記直交座標系上での位置偏差から計算される前記直交座標系上での力を逆動力学で解いて求められた各軸のトルク T_i ($i = 1, 2 \dots n$; 但し、 n はロボットの軸数)

と、前記位置ループ処理のゲイン K_p と、速度ループ処理のゲイン K_v に基づいて算出された E_i を、各軸の前記位置ループ処理の入力とすると共に、

前記位置ループ処理の出力を予め定められた積分ゲインで積分し、前記速度ループ処理の出力に予め定められた所定の制限内で足し込んでトルク指令を出力するようにした、直交座標系上で柔らかさが調節可能な前記サーボ系。

【請求項 4】 ロボットの各軸を駆動するサーボモータを制御する為に前記各軸毎に位置ループ処理と速度ループ処理を含む処理を行なうサーボ系であって、

作業空間内に定義された直交座標系上での前記サーボの柔らかさを表わすパラメータの設定値の組と前記直交座標系上での位置偏差から計算される前記直交座標系上での力を逆動力学で解いて求められた各軸のトルク T_i ($i = 1, 2 \dots n$; 但し、 n はロボットの軸数) を

前記位置ループ処理のゲイン K_p と速度ループ処理のゲ

イン K_v の積で除した値 $E_i = T_i / (K_p K_v)$ を、各軸の前記位置ループ処理の入力とすると共に、前記位置ループ処理の出力を予め定められた積分ゲインで積分し、前記速度ループ処理の出力に予め定められた所定の制限内で足し込んでトルク指令を出力するようにした、直交座標系上で柔らかさが調節可能な前記サーボ系。

【請求項 5】 前記作業空間内に定義された直交座標系上での前記サーボの柔らかさを表わすパラメータの組が、画面入力により設定される請求項 1 ～請求項 4 いずれか 1 項に記載された、直交座標系上で柔らかさが調節可能な前記サーボ系。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本願発明は産業用ロボット（以下、単に「ロボット」と言う。）の制御技術に関し、更に詳しく言えば、ロボット軸駆動用のサーボモータを制御するサーボ系の柔らかさを必要に応じて変更し、可変に指定乃至設定すること（以下、「調節」と言う。）が出来るようにする為の技術に関する。

【0002】

【従来の技術】 ロボットのアームを駆動する各軸のサーボモータの制御は、通常、位置制御ループ及び速度制御ループを有するサーボ系によって制御されている。図 1 は、これを示したブロック図で、符号 1 は位置ループゲイン K_p の項、符号 2 は速度ループゲイン K_v の項である。また符号 3、4 はモータの伝達関数の項であり、3 はトルク定数 K_t 、4 はイナーシャ J である。更に、符号 5 はモータ速度 v を積分してモータ位置 q を求める伝達関数である。なお、 s はラプラス演算子を表わしている。

【0003】 ロボットコントローラの内部で作成される移動指令 r とモータ位置 q より位置偏差 e が算出され、該位置偏差 e に位置ループゲイン K_p を乗じて速度指令 v_c が出力される。更に、速度偏差 e_v が速度指令 v_c とモータ速度 v より算出され、この速度偏差 e に速度ループゲイン K_v を乗じてトルク指令 t_c が出力される。そして、該トルク指令 t_c に応じた駆動電流がモータに供給される。なお、速度ループの制御においては、 P 制御に代えて PI 制御または IP 制御が適用されることもある。

【0004】 このようなサーボ系（ PI 制御もしくは IP 制御）によって各軸のモータが制御されるロボットのツール先端点が目標位置に向かって移動中になんらかの障害物に遭遇・接触した場合、ロボットのツール先端点は障害物に抗して目標位置に向かって移動を続行しようとする現象が生じる。

【0005】 この現象は次のように説明される。即ち、障害物の存在に関わらずサーボモータは目標位置に向かって移動しようとするが、実際には障害物によりその目

標位置に移動することが阻まれ位置偏差 e が増大する。その結果、この位置偏差に位置ゲイン K_p を乗じて得られる速度指令 v_c も増大する。そして、この増大する速度指令 v_c とモータの速度 v (障害物に当接しているときには、速度 v は「0」に近いと考えられる。)との差は、速度ループの持つ積分器によって積分されて増大し、トルク指令 T_c は大きな値となる。

【0006】結局、サーボモータは目標位置への移動を実現しようとして最大のトルクを出力するようになり、ロボットの停止やワーク、エンドエフェクタ等の破壊事故(干渉事故)を招く原因となる。そこで、このような不都合を回避する手段として、必要に応じて位置ループゲイン K_p 及び速度ループゲイン K_v を低下させることにより速度指令 v_c 及びトルク指令 t_c の値の増大を抑止する方式が用いられている。

【0007】この方式においては、通常、柔軟制御用のゲイン値 K_p' 、 K_v' が予め設定され、柔軟制御指令の入力時にゲインの値が K_p 、 K_v から K_p' 、 K_v' に各々切り換えられる。

【0008】このような従来方式は、サーボ系の柔らかさを各ロボット軸(以下、単に「各軸」とも言う。)毎に調節することを通して「各軸空間上でのソフトフロート機能」を実現するものである。

【0009】例えば、ロボットの手首側の3軸について非常に小さな K_p' 、 K_v' を設定し、手首側の3軸について特に柔らかいサーボ制御が実現される。また、全軸について非常に小さな K_p' 、 K_v' を設定すれば、全軸について非常に柔らかいサーボ系が構成される。

【0010】

【発明が解決しようとする課題】しかし、上記した従来方式では、サーボ系の柔らかさは各軸上でしか調節することが出来ないで、作業空間内の方向別に柔らかさを調節し、嵌め合い、バリ取り加工等、実際の作業に適した柔らかさをロボットに持たせることが困難であった。

【0011】例えば、嵌め合いにおいては、一般に、ロボットに把持されたワークの挿入方向に関しては硬く、挿入方向と直交する方向に関しては柔らかいサーボ系による制御が行なわれることが望ましいと考えられが、作業空間内に定義された座標系上でサーボ系の柔らかさを調節出来ない従来方式ではこのようなニーズには応えられない。

【0012】また、バリ取り加工においては、加工線の延在方向とそれに直交する2方向の動作について各々別の柔らかさを持たせることが望ましいが、それをロボットの各軸についてサーボ系の柔らかさを調節する従来方式で実現することも極めて困難である。

【0013】そこで本願発明の目的は、作業空間内に定*

$$K_x (x_d - x) = F_x \quad \dots (1)$$

$$K_y (y_d - y) = F_y \quad \dots (2)$$

*義された直交座標系(以下、単に「直交座標系」とも言う。)上で柔らかさを調節することが出来るサーボ系を提供し、ロボットに直交座標系上でのソフトフロート機能を持たせることにある。また、本願発明は、そのことを通して、ロボットを用いた作業の精度と安全性を向上させることを企図するものである。

【0014】

【課題を解決するための手段】本願発明は、ロボットの各軸を駆動するサーボモータを制御する為に各軸毎に位置ループ処理と速度ループ処理を含む処理を行なうサーボ系に、直交座標系上でソフトフロート機能を持たせるようにしたものである。先ず、作業空間内に定義された直交座標系上でのサーボの柔らかさを表わすパラメータの組を設定する。設定は、例えばロボットコントローラに付属した教示操作盤から画面入力することで行なわれる。ソフトフロート機能を有効化した場合には、前記設定されたパラメータの組と、直交座標系上での位置偏差から直交座標系上での力を計算する。

【0015】そして、この力を逆動力学で解いて各軸のトルク T_i ($i=1, 2, \dots, n$; 但し、 n はロボットの軸数)を求め、このトルク T_i と位置ループ処理のゲイン K_p と、速度ループ処理のゲイン K_v に基づいて算出された E_i を、各軸の前記位置ループ処理の入力とした処理を行なうことにより、直交座標系上で指定された柔らかさが実現される。柔らかさの調節は、パラメータの組の設定を通して実行される。位置ループ処理の入力 E_i は、前記トルク T_i ($i=1, 2, \dots, n$; 但し、 n はロボットの軸数)を前記位置ループ処理のゲイン K_p と速度ループ処理のゲイン K_v の積で除すことによって算出することが出来る。

【0016】重力の影響を考慮する必要がある場合には、位置ループ処理の出力が予め定められた積分ゲインで積分され、速度ループ処理の出力に足し込まれてトルク指令が出力される。その際の足し込み量には一定の制限を設けることで、柔らかさが失われることを防止する。

【0017】

【作用】本願発明は、ロボットの各軸を駆動するサーボモータを制御するサーボ系についてソフトフロート機能を拡充するものである。即ち、本願発明によれば、直交座標系上で柔らかさを指定することが出来るサーボ系が構築され、「直交座標系上でのソフトフロート機能」が実現される。本願発明の原理の概要を説明すれば、次の通りである。

【0018】直交座標系(O-X-Y-Z)上でのロボット動作制御の柔らかさは、次のように表わすことが出来る。

$$K_z (z_d - z) = F_z$$

ここで、

K_x, K_y, K_z : X, Y, Z 各座標系軸方向に関する
柔らかさの程度を表わすパラメータ

x, y, z : 直交座標系上でのフィードバック (X, Y, Z 各成分)

x_d, y_d, z_d : 直交座標系上での位置指令 (X, Y, Z 各成分)

F_x, F_y, F_z : ツール先端点にかかる力 (位置指令
で指定された位置に移動しようとする力の X, Y, Z 各
成分)、

である。
【0019】従って、直交座標系上でソフトフロート機能
を実現する為には、ロボットの動作時に上記 (1) ~
(3) 式中のパラメータ K_x, K_y, K_z の値を (可変
に定められた) 一定値に保つような各軸の柔軟制御を行
なうサーボ系を構築すれば良いことになる。

【0020】上記 (1) ~ (3) 式は、ツール先端点に
かかる力 F_x, F_y, F_z が、パラメータ K_x, K_y, K_z と、直交座標系上における位置偏差で表わされるこ
ことを示している。

6

... (3)

【0021】一方、逆動力学のアプローチとして良く
知られている Newton-Euler 法 (Luh のア
プローチとも呼ばれる。) を用いれば、ツール先端点
にかかる力 F_x, F_y, F_z から、ロボットの各軸に必
要なトルク τ_i ($i=1, 2, \dots, n$; n はロボットの
軸数、以下同様。) を求めることが出来る。

【0022】Newton-Euler 法は、例えば
「機械系のためのロボティクス」 (遠山茂樹 著; 総合
電子出版社、1991年5月30日第2版発行; p55
~ p65) に詳しいのでその詳細は省略し、使用される
アプローチの結論部分のみを記せば次のようになる。
このアプローチは、初期化、フォワードルーチン、バ
ックワードルーチンで構成される。

【0023】フォワードルーチンでは、ベース座標系 Σ_0
側からロボット手先側に向かって各リンクの運動学的
な情報が計算される。一方、バックルーチンでは、ロボ
ット手先側からベース座標系 Σ_0 側へ向かって各リンク
に作用する力/モーメントが計算される。

【0024】

【数1】

[ステップ0] 初期化

$$\omega_0 = \dot{\omega}_0 = 0; \dot{v}_0 = g = (g_x, g_y, g_z) \quad z_0 = (0, 0, 1)$$

[ステップ1] フォワードルーチン ($i = 1, 2, \dots, n$)

$$\omega_i^{(1)} = R_i^T (\omega_{i-1}^{(1-1)} + z_0 \dot{q}_i) \quad (\text{回転軸の場合})$$

$$\omega_i^{(1)} = R_i^T \omega_{i-1}^{(1-1)} \quad (\text{直動軸の場合})$$

$$\dot{\omega}_i^{(1)} = R_i^T (\dot{\omega}_{i-1}^{(1-1)} + z_0 \ddot{q}_i + \omega_{i-1}^{(1-1)} \times z_0 \dot{q}_i) \quad (\text{回転軸の場合})$$

$$\dot{\omega}_i^{(1)} = R_i^T \dot{\omega}_{i-1}^{(1-1)} \quad (\text{直動軸の場合})$$

$$\dot{v}_i^{(1)} = \omega_i^{(1)} \times p_i^* + \omega_i^{(1)} \times (\omega_i^{(1)} \times p_i^*) + R_i^T \dot{v}_{i-1}^{(1-1)} \quad (\text{回転軸の場合})$$

$$\dot{v}_i^{(1)} = R_i^T (z_0 \ddot{q}_i + \dot{v}_{i-1}^{(1-1)}) + \omega_i^{(1)} \times p_i^* + 2\omega_i^{(1)} \times (R_i^T z_0 \dot{q}_i) + \omega_i^{(1)} \times (\omega_i^{(1)} \times p_i^*) \quad (\text{直動軸の場合})$$

$$\dot{\bar{v}}_i^{(1)} = \omega_i^{(1)} \times \bar{s}_i^{(1)} + \omega_i^{(1)} \times (\omega_i^{(1)} \times \bar{s}_i^{(1)}) + \dot{v}_i^{(1)}$$

$$F_i^{(1)} = m_i \times \dot{\bar{v}}_i^{(1)} \quad N_i = I_i \times \omega_i^{(1)} + \omega_i^{(1)} \times I_i \omega_i^{(1)}$$

[ステップ2] バックワードルーチン ($i = n, n-1, \dots, 1$)

$$f_i^{(1)} = R_{i+1}^T f_{i+1}^{(1+1)} + F_i^{(1)}$$

$$n_i^{(1)} = R_{i+1}^T n_{i+1}^{(1+1)} + p_i^* \times R_{i+1}^T f_{i+1}^{(1+1)} + (p_i^* + \bar{s}_i^{(1)}) \times F_i^{(1)} + N_i^{(1)}$$

$$\tau_i = n_i^{(1)} R_i^T z_0 + b_i \dot{q}_i \quad (\text{回転軸の場合})$$

$$\tau_i = f_i^{(1)} R_i^T z_0 + b_i \dot{q}_i \quad (\text{直動軸の場合})$$

各記号の意味は次の通りである。

$n_i^{(1)}$; $i-1$ 番目の軸で駆動されるリンク (以下、リンク $i-1$ などと言う。) からリンク i に及ぼされるトルクの Σi 表示。但し、 Σi は i 番目の軸上に設定された座標系 Σi を表わす。

$f_i^{(1)}$; リンク $i-1$ からリンク i に及ぼされる力の Σi 表示

$F_i^{(1)}$; リンク i に及ぼされる外力ベクトルの Σi 表示

$N_i^{(1)}$; リンク i に及ぼされるモーメントベクトルの Σi 表示

R_i ; 座標系変換行列 A_i の主座の 3×3 行列で、回転を表わす。なお、座標系変換行列 A_i は、 i 番目の軸上に設定された座標系 Σi から $i-1$ 番目の軸上に設定された座標系 $\Sigma i-1$ への座標変換行列である。但し、座標系 $\Sigma 0$ は前記直交座標系と一致させたベース座標系にとる。

【0025】 p_i^* ; Σi 表示の $\Sigma i-1$ 原点から Σi 原点への位置ベクトル

g ; 重力ベクトル

s_i バー ; Σi 表示のリンク i の重心位置を示す位置ベクトル

I_i ; Σi 表示のリンク i の重心周りの慣性モーメント

z_0 ; $\Sigma 0$ の Z 軸方向の単位ベクトルで、 $\Sigma 0$ 上では $(0, 0, 1)$ と表わされる

b_i ; i 番目の関節軸の粘性抵抗

q_i ; i 番目の関節軸の軸変数で、 $q_i = \theta_i$ (リンク i の角度) として良い

ω, ω ドット ; $\Sigma 0$ 表示のリンク i の角速度、角加速度
上記アルゴリズムにおいて使用される諸量は、ロボット
の構造パラメータ、各軸の位置、速度、加速度等から計算される。また、前述の式 (1) ~ (3) に従って位置
偏差 $(x_d - x)$, $(y_d - y)$, $(z_d - z)$ と柔らかさを表わすパラメータ K_x, K_y, K_z から定められる F_x, F_y, F_z は、上記アルゴリズム中の $f_{n+1}^{(n+1)}$ に取入れられる。 $f_{n+1}^{(n+1)}$ は、 $\Sigma n+1$ をツール座標系として定義することにより、ツール座標系の原

点に働く力をツール座標系上で表わしていると考えることが出来る。

【0026】そこで、ツール座標系の姿勢を表わす現在データを行列 U とし、 $F = (F_x, F_y, F_z)^T = (F_x^{(0)}, F_y^{(0)}, F_z^{(0)})^T$ とすれば、行列演算 UF によって、 F_x, F_y, F_z のツール座標系上の表現が計算される。これを $F^{(n+1)} = (F_x^{(n+1)}, F_y^{(n+1)},$ *

$$K_p K_v E_i = T_i$$

但し、 K_p は位置ループゲイン、 K_v は速度ループゲイン、 E_i は軸 i の位置ループの入力を表わしている。図2は、このサーボ系を図1に準じた表記で示したものである(K_p, K_v は便宜上各軸共通の表記としたが、各軸毎に設定される場合には K_p, K_v に代えて K_{pi}, K_{vi} とすれば良い)。図1と図2の差異は位置ループ処理への入力 e から E_i に変わっていることである。即ち、各軸の位置ループ入力 E_i は従来のサーボ系(図1参照)における位置偏差 e のように、その軸 i の位置指令 r と位置フィードバック q の偏差で定義されるものではないことである。

【0029】この E_i は、直交座標系上でのサーボの柔※20

$$E = T / (K_p K_v)$$

以上のことから、各軸について上記(6)式で決定される E_i を各軸の位置ループ入力とすることにより、前述の関係式(1)～(3)で表わされる直交座標系上での柔らかさを実現するために必要なトルク T_i が発生する。

【0031】次に重力の作用を考慮する。重力の作用を考慮に入れた制御は、図2に示したサーボ系に積分器を付加することで達成出来る。これを、図2に準じた表記で図3に示す。図3において、 K_p, K_v は便宜上各軸共通の表記としたが、各軸毎に設定される場合には K_p, K_v に代えて K_{pi}, K_{vi} とすれば良いことは図2の場合と同様である。図2と図3の差異は、 E_i を入力とする位置ループ処理で出力される速度指令 v_c に対して重力分を補償する積分項6(K_g / s ;但し、 K_g は積分器のゲイン)が計算され、速度指令 v_c を入力とする速度ループ処理で計算されるトルク T_i に足し込まれることである。

【0032】この積分項6は、鉛直上向き方向についてトルク不足となる状態を回避する手段を提供するもので★

$$\text{IF } I > I_{GR} + I_{SOFT-LIM} \text{ THEN}$$

$$\text{IF } I < I_{GR} - I_{SOFT-LIM} \text{ THEN}$$

以上説明した手法に従って各軸のサーボ系を構成することによって、ロボットに直交座標系上で指定された柔らかさを持たせることが出来る。例えば、Z軸方向を挿入方向とする嵌め合いにおいては、 K_x と K_y は比較的小さく、 K_z は比較的大きく指定することで、嵌め合いをスムーズに行なうことが出来る。また、バリ取り加工においては、加工線の延在方向(例えばX軸方向)とそれに直交する2方向(例えばY軸、Z軸方向)について、個別に適正な K_x, K_y, K_z の値を設定し、バリ取り

* $F_z^{(n+1)})^T$ とすれば、 $f_{n+1}^{(n+1)} = F^{(n+1)}$ となる。

【0027】ここで、重力を考慮しない条件; $g = (0, 0, 0)$ で上記アルゴリズムを用いて求められた軸 i のトルク τ_i ($g = 0$)を T_i とすれば、その軸のサーボ系は次のように表わされる。

【0028】

$$\dots (5)$$

※らかさを表わすパラメータの組 K_x, K_y, K_z と直交座標系上での位置偏差($x_d - x$), ($y_d - y$), ($z_d - z$)から計算される力を逆動力学で解いて求められた各軸のトルク T_i と、位置ループゲイン K_p と、速度ループゲイン K_v に基づいて算出されたものである。

【0030】換言すれば、この E_i は、直交座標系上でパラメータ K_x, K_y, K_z で表わされる柔らかさを実現させる為にそのロボット軸のサーボ系が持つべき柔らかさに応じて各軸の位置偏差 e を修正したものと解釈することも出来る。 E を次式(6)で記す。

$$\dots (6)$$

★ある。例えば直交座標系のZ軸を鉛直上向き方向にとり、 K_z を小さく(柔らかさは大)に設定した場合、積分項6がない場合には T_i だけでは重力に打ち勝つことさえ出来ず、落下事故を起こす可能性がある。積分項6があれば、 T_i だけでは重力に打ち勝つことが出来なくとも、ゲイン K_g に応じてトルク T_i' が増大し、十分なトルクが発生する。

【0033】但し、この積分項6で表わされる積分器には重力分のトルクが溜まる性質があるので、積分器の出力に一定の制限値を設ける必要がある。もし、制限値を設けないと、積分器に外力以上の力が溜り、柔らかさを失ってその軸の指令位置に向かう強制的な動きが発生する。

【0034】そこで、ソフトフロート機能起動時に積分器に溜っている値(出力)を I_{GR} として、ソフトフロート機能時には次のような制限を設ける。 $I_{SOFT-LIM}$ の値はチューニング等によって各軸毎に適当に設定される。

$$I = I_{GR} + I_{SOFT-LIM} \dots (7)$$

$$I = I_{GR} - I_{SOFT-LIM} \dots (8)$$

加工の作業精度を向上させることが出来る。

【0035】

【実施例】図4は、本願発明の実施例で使用されるロボットコントローラRCを関連機器と共に要部ブロック図で示したものである。まず、ソフトフローティングサーボ制御に直接関係のある部分から説明すると、符号10はシステム全体を制御するホストコンピュータを表わしている。符号16は、ホストコンピュータ10から出力される移動指令や制御信号を後述のデジタルサーボ回

路のプロセッサに引渡し、あるいは、逆にデジタルサーボ回路のプロセッサからの各種信号をホストコンピュータ 10 に引き渡すための共有 RAM メモリを表わしている。

【0036】符号 17 は、上述したサーボ制御を実行するデジタルサーボ（ソフトウェアサーボ）回路で、プロセッサ、ROM、RAM 等のメモリ等で構成される。符号 18 は、ロボット 30 における各軸のサーボモータの位置のフィードバック値、速度のフィードバック値、電流のフィードバック値等が書き込まれる帰還レジスタを表わしている。

【0037】他の部分について見ると、ホスト CPU 10 にはバスライン 19 を介して ROM 11、RAM 12、不揮発性メモリ 13、外部装置 40 とのインターフェースの役割を果たす入出力装置（I/O）14、教示操作盤 20 とのインターフェイス（I/F）15 が接続されている。ROM 11 には、各種のシステムプログラムが格納される。RAM 12 は、ホスト CPU 10 によるデータの一時記憶に使用されるメモリである。不揮発性メモリ 13 には、ロボット 30 及び外部装置 40 の動作内容に関する各種プログラム、関連設定値等が格納される。

【0038】教示操作盤 20 は、液晶ディスプレイ（LCD）及びキーボード KB を備え、プログラム再生運転指令、ジョグ送り指令、プログラムデータの入力／変更、関連設定値入力／変更等が可能である。本実施例では、後述する柔らかさ設定画面で画面入力される XYZ 各軸方向の柔らかさのデータ、並びに教示操作盤 20 から入力される ISOFT-LIM の値が不揮発性メモリ 13 に格納される。

【0039】入出力装置 14 に接続される外部装置は、アプリケーションによって異なる。例えば、ワークの把持を伴う作業であればロボットハンド、バリ取り加工であればバリ取りツール（電源等を含む。）が入出力装置 14 に接続される。

【0040】図 5 は、本願発明の適用例として考えられる嵌め合い作業について説明する図である。同図において、符号 W1、W2 は嵌め合い対象のワークを表わしている。ワーク W1 は図示を省略したロボットハンドに把持されて、ワーク W2 の孔 H L 内に嵌入される。嵌入方向はロボットコントローラ RC に設定済みの直交座標系 $\Sigma 0$ の Z 軸方向と一致しているものとする。符号 GP はワーク W1 の先端部に設けられたガイドピンを表わし、TCP はツール先端点を表わす。

【0041】また、P1、P2 は各々後述する動作プログラムでイチ [1] 及びイチ [2] として教示された位置を表わす。このような嵌め合いにおいては、教示ライン P1 P2 が孔 H L の軸線とある程度ずれることが多い。従って、イチ [1] からイチ [2] への移動時に通常の硬いサーボ制御を行なうと、TCP はあくまで教示

ライン P1 P2 に沿って進もうとするので、ガイドピン GP が孔 H L の入口でつかえて動けなくなり、場合によってはワーク W1、W2 やロボットハンドの破損事故を起こす。本願発明を適用すれば、このような事態が回避される。以下、その為の手順を説明する。

【0042】まず、ユーザは、教示操作盤 20 に付設された LCD に図 6 に示されたような柔らかさ入力画面を呼び出す。そして、 $\Sigma 0$ の X 軸、Y 軸、Z 軸方向の各々について個別に所望の値を画面入力する。入力される数値が小さい程柔らかい状態が設定され、入力される数値が大きい程硬い状態が設定される。ここでは、X 軸方向： $K_x = 200 \text{ g/cm}$ 、Y 軸方向： $K_y = 200 \text{ g/cm}$ 、Z 軸方向： $K_z = 500 \text{ g/cm}$ が各々設定される例が示されている。

【0043】図 7 は、図 5 に示した嵌め合いを実行する為の直交座標系ソフトフロート機能を用いた動作プログラムの一例を表わしている。

【0044】図 6 の画面入力後に図 7 の動作プログラムの再生運転を開始すると、ロボットのツール先端点はイチ [1] まで各軸移動形式で移動し、位置決めされる。そして、ハンドを閉じてワーク W1 を把持した後に、直交座標系ソフトフロート機能を有効にして、イチ [2] に向かって指令速度 50 mm/sec で移動する。直交座標系ソフトフロート機能が有効にされると図 8 に示した処理が所定周期で実行され、作用の欄で説明した原理によって、ソフトフロート機能が発揮される。まず、ステップ S1 では、直交座標系 $\Sigma 0$ 上における位置偏差と設定されたパラメータ (K_x , K_y , K_z) を用いて、直交座標系 $\Sigma 0$ 上における力 (F_x , F_y , F_z) を求める。直交座標系 $\Sigma 0$ 上における位置偏差は、各軸における位置偏差から順運動学で計算出来る。続くステップ S2 では、直交座標系 $\Sigma 0$ 上で求められた力 (F_x , F_y , F_z) をロボットの姿勢データを用いてツール座標系 Σ_{n+1} 上のデータ ($F_x^{(n+1)}$, $F_y^{(n+1)}$, $F_z^{(n+1)}$) に変換する。

【0045】更に、($F_x^{(n+1)}$, $F_y^{(n+1)}$, $F_z^{(n+1)}$) = ($f_x^{(n+1)}$, $f_y^{(n+1)}$, $f_z^{(n+1)}$) として、作用の欄で説明した Newton-Euler 法により、各軸のトルク T_i を計算し（ステップ S3）、 $T_i / (K_p K_v)$ から位置ループ入力 E_i を計算する（ステップ S4）。計算された位置ループ入力 E_i を入力とする位置ループ処理、位置ループ出力を入力とする積分処理と速度ループ処理、積分処理と速度ループ処理の出力を加算するトルク指令作成処理等を実行すれば（ステップ S5）、図 6 の設定画面通りのソフトフロート機能が発揮される。積分処理と速度ループ処理の出力の加算は、設定された ISOFT-LIM の下で、前出の (7)、(8) 式の制限を守った範囲で行なわれる。

【0046】これにより、ワーク W1 のガイドピン GP が孔 H L の上縁部に当接すると、ツール先端点 TCP は

孔H Lの内壁をなぞるように孔H Lの軸線W J上の点Q (教示位置P2に対応する位置)まで円滑に移動する。次いで、ロボットハンドを開放して、ワークW1の把持を解けば嵌め合い作業は完了する。

【0047】なお、上記実施例では、直交座標系上で柔らかさを表わすパラメータの設定は画面入力で行なったが、動作プログラムの中で指定する方式としても良い。例えば、バリ取りへのアプリケーションにおいて、加工線の延在方向が屈曲している場合に、動作プログラム中でそれに応じたパラメータ変更を行なえば、加工線の延

【0048】

【発明の効果】本願発明によれば、作業空間内に定義された直交座標系上で柔らかさを調節することが出来る。本願発明の手法に従って各軸のサーボ系を構成することによって、例えば、嵌め合い、バリ取り等の作業精度を向上させることが出来る。

【図面の簡単な説明】

【図1】従来のサーボ系の構成を示したブロック図である。

【図2】重力を考慮しない場合の本願発明のサーボ系の構成を示したブロック図である。

【図3】重力を考慮した場合の本願発明のサーボ系の構成を示したブロック図である。

【図4】本願発明の実施例で使用するロボットコントローラの概要を説明する要部ブロック図である。

【図5】本願発明の適用例として考えられる嵌め合い作業について説明する図である。

【図6】直交座標系上で柔らかさを入力する為の画面を説明する図である。

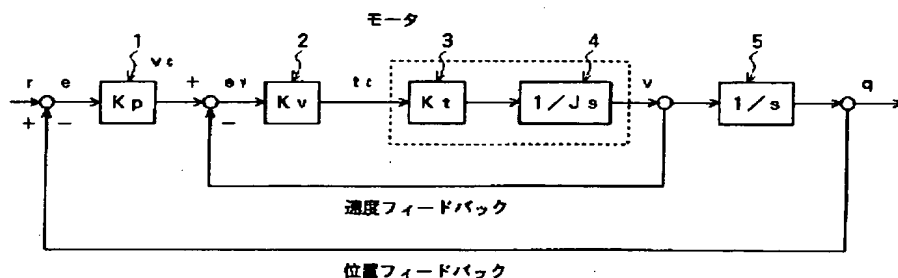
【図7】図5に示した嵌め合いを実行する為に直交座標系ソフトフロート機能を用いた動作プログラムの一例を表わしたものである。

【図8】実施例におけるソフトフロート機能有効化時の処理の概要を説明するフローチャートである。

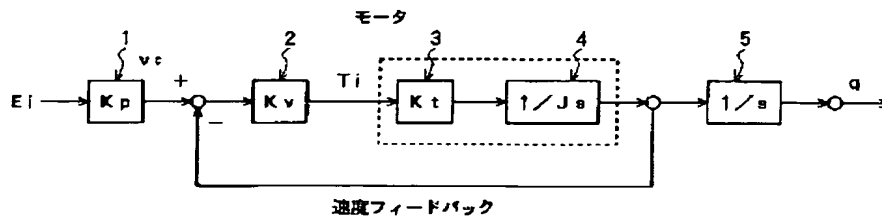
【符号の説明】

- 1 位置ループゲインの項
- 2 速度ループゲインの項
- 3, 4 モータの伝達関数の項
- 5 モータ速度からモータ位置を求める伝達関数
- 6 重力分を補償する積分項
- 10 ホストCPU
- 11 ROM
- 12 RAM
- 13 不揮発性メモリ
- 14 入出力装置
- 15 インターフェイス
- 16 共有RAM
- 17 デジタルサーボ回路
- 18 帰還レジスタ
- 19 バスライン
- 20 教示操作盤
- 30 ロボット
- 40 外部装置
- KB キーボード
- Kp 位置ゲイン
- Kv 速度ループゲイン
- Kt トルク定数
- Kg 積分ゲイン
- GP ガイドピン
- HL 孔
- LCD 液晶ディスプレイ
- P1, P2 教示点
- RC ロボットコントローラ
- W1, W2 嵌め合い対象ワーク
- WJ 孔の軸線

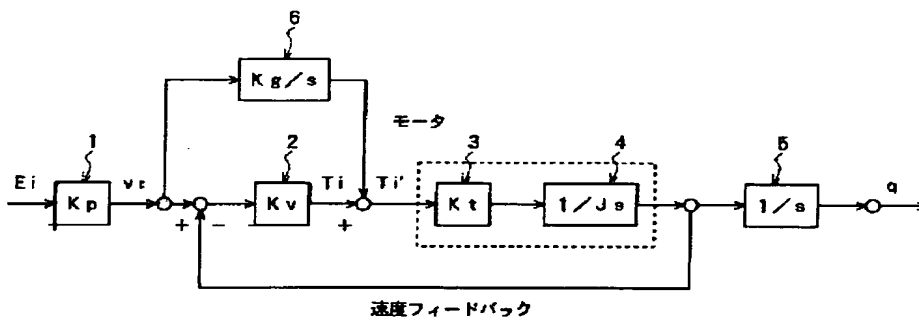
【図1】



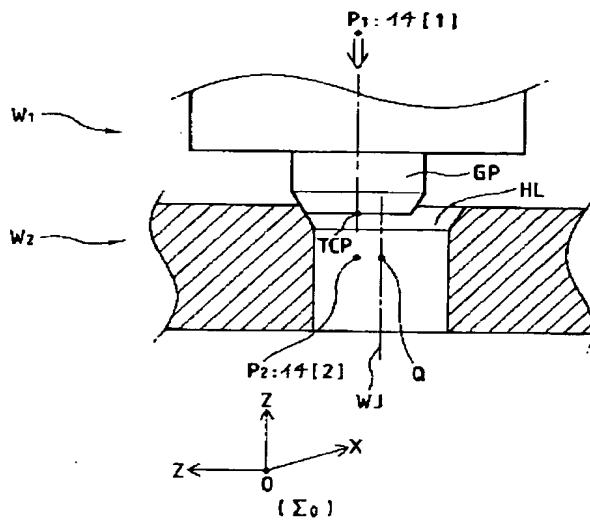
【図2】



【図3】



【図5】



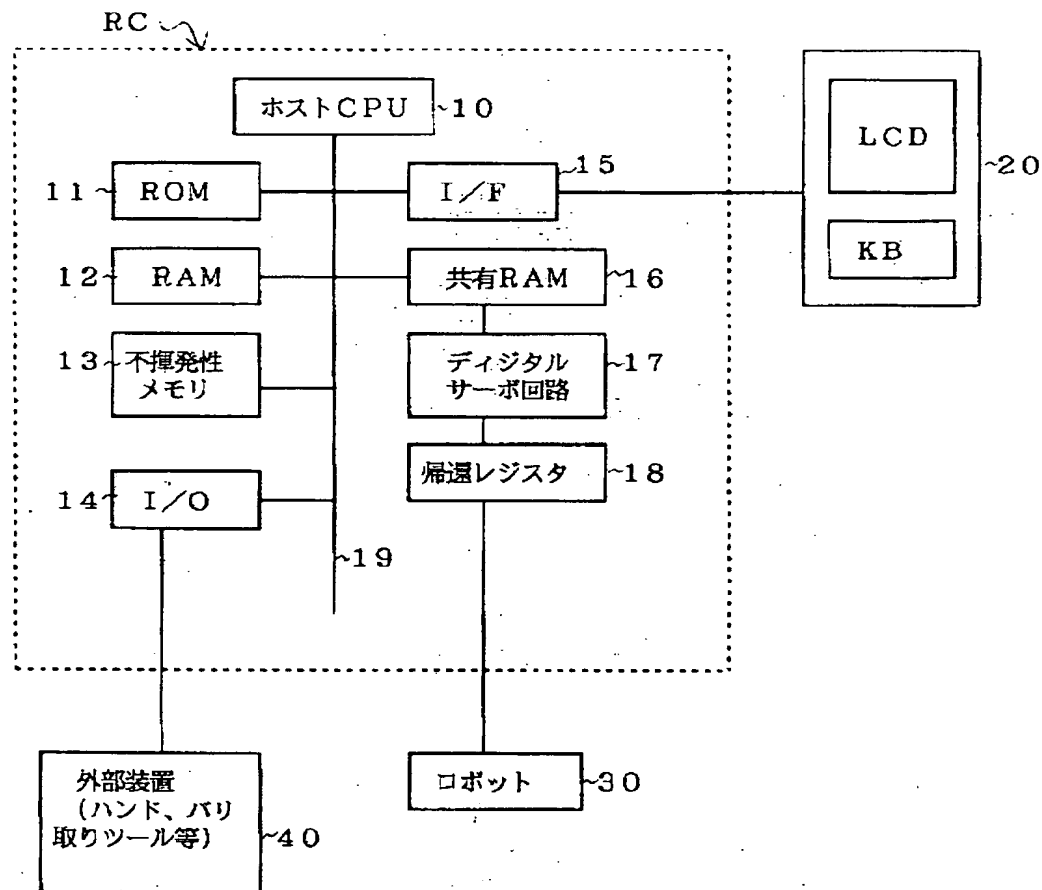
【図6】

セッティ/チェックソフトウェアソフト	
1 Xボウコウヤワラカサ	: 200 g/cm
2 Yボウコウヤワラカサ	: 200 g/cm
3 Zボウコウヤワラカサ	: 500 g/cm

【図7】

プログラム ハメアイ	
1:	カクジク イチ [1] 100% イチギ
2:	ハンド トジル
3:	チェックウザヒウソフトフロート カイシ
4:	チェックセン イチ [2] 50mm/sec
5:	ハンド ヒラク
6:	チェックウザヒウソフトフロート オウリ

【図4】



【図8】

